

TARTU ÜLIKOOL
Arvutiteaduse instituut
Informaatika õppekava

Vello Vaherpuu
Murelahendajate loomise keskkond
Bakalaureusetöö (9 EAP)

Juhendaja: Eno Tõnisson, MSc

Tartu 2016

Murelahendajate loomise keskkond

Lühikokkuvõte:

Käesoleva bakalaureusetöö kirjutamise käigus valmis murelahendajate ehk samm-sammuliste tõrkeotsijate (ingl troubleshooter) loomise keskkond, mis on juba kasutust leidnud Tartu Ülikooli arvutiteaduse instituudi korraldatud MOOCides (vaba juurdepääsuga e-kursus). Keskkonna loomisel kasutati järgmiseid tehnoloogiaid: CSS, HTML, JavaScript, AngularJs, D3.js ja MongoDB. Keskkonna kasutamiseks on vaja luua rakendusse kasutajakonto, mis võimaldab kasutajal luua, muuta ning jagada murelahendajaid, samuti on võimalik vaadata nende kohta statistilisi andmeid. Rakenduse loomisel on peetud silmas eeskätt MOOCidest tulenevaid vajadusi, kuid on mõeldud kasutamiseks ka väljaspool MOOCe.

Võtmesõnad:

MOOC, murelahendaja, veebirakendus

CERCS: P175, informaatika, süsteemiteooria

Environment for creating step-by-step troubleshooters

Abstract:

In the process of writing this Bachelor's Thesis a web application for creating and sharing step-by-step troubleshooters was created. It has already been used in several MOOCs (Massive Open Online Course) held by the University of Tartu. Technologies used for creating the web application were CSS, HTML, JavaScript, AngularJS, D3.js and MongoDB. Registered user can create, modify and share troubleshooters in addition application allows insights how visitors use each guide. Most of the functionality is derived from the requirements of MOOCs but it's intended to be used outside of the MOOC as well.

Keywords:

MOOC, troubleshooter, troubleshooting, web application

CERCS: P175, informatics, systems theory

Sisukord

1. Sissejuhatus.....	5
2. Ülevaade	6
2.1 Tõrkeotsing	6
2.2 Loodava keskkonnaga sarnased keskkonnad.....	6
2.2.1 Hinnavõrdlus ühe kuu raames.....	7
2.2.2 Kokkuvõte.....	7
2.3 Tõrkeotsingu näited	7
2.3.1 Telia	8
2.3.2 Google.....	8
2.3.3 YouMail	9
2.3.4 Kokkuvõte.....	9
3. Keskkonna kirjeldus	10
3.1 Rollid	10
3.2 Kuidas külastaja antud keskkonda kasutab?.....	10
3.3 Mittefunktsionaalsed nõuded	10
3.3.1 Toetatavus	10
3.3.2 Jõudlus	10
3.3.3 Käideldavus.....	10
3.3.4 Kasutatavus	11
3.4 Funktsionaalsus.....	11
3.4.1 Registreerimine	11
3.4.2 Sisselogimine	11
3.4.3 Parooli taastamine	12
3.4.4 Murelahendaja loomine	12
3.4.5 Murelahendajate kuvamine	13
3.4.6 Murelahendajate seadete muutmine.....	14
3.4.7 Murelahendaja kustutamine	14
3.4.8 Murelahendaja jagamine keskkonnasiseselt	14
3.4.9 Murelahendaja jagamine väljaspool keskkonda	14
3.4.10 Murelahendaja kloonimine	14
3.4.11 Murelahendaja sisu loomine	14
3.4.11.1 Murelahendaja sisu loomise vaade	14
3.4.11.2 Murelahendaja sammude loomine	15

3.4.11.3 Murelahendaja sammude kustutamine.....	15
3.4.11.4 Murelahendaja sammude järjekorra muutmine	16
3.4.11.5 Murelahendaja seadete muutmine sisu loomise vaates.....	16
3.4.11.6 Murelahendaja sammu sisu loomine.....	16
3.4.12 Murelahendaja statistika	18
4. Kasutatud tehnoloogiad	20
4.1 Disain	20
4.2 Veebirakenduse serveripoolne osa.....	20
4.2.1 Moodulid.....	20
4.2.2 Andmebaas.....	21
4.2.3 E-mailide saatmine.....	21
4.3 Kliendipoolne osa	22
4.3.1 AngularJs	22
4.3.2 Graafikute kuvamine.....	22
5. Tulemus	23
5.1 Kasutegur	23
5.2 Tagasiside	23
6. Kokkuvõte.....	24
7. Viited	25
I. Rakenduse üles seadmine Linux masinal	27
II. Demo video.....	27
III. Litsents.....	28

1. Sissejuhatus

Kuna praegusel ajal on vaba juurdepääsuga e-kursuste ehk MOOCide (Massive Open Online Course) [1] populaarsus järjest kasvamas, on ka Tartu Ülikooli arvutiteaduse instituut loonud praeguseks 2 MOOCi: “Programmeerimine maalähedaselt” ja “Programmeerimise alused”. Kursus “Programmeerimine maalähedaselt” (1 EAP) on mõeldud peamiselt inimestele, kellel puudub eelnev kokkupuude programmeerimisega. Esmakordselt toimus see 1. detsember – 28. detsember 2014 ning praeguseks on seda läbi viidud 4 korda. Kursus “Programmeerimise alused” (3 EAP) on võrdväärne samanimelise statsionaarse ainega. Pilootkursus toimus 11. jaanuar - 6. märts 2016. Töö kirjutamise ajal algas kursus ka teist korda, toimumise ajaks on märgitud 28. märts – 22. mai 2016.

Kursuste loomise ja korraldamisega tegeles peale antud töö autori ka palju teisi inimesi. Seega edaspidi kutsutakse töö autorit ja teisi tiimi liikmeid ühiselt kursuse korraldajateks.

Kursuse materjalid asuvad Tartu Ülikooli arvutiteaduse instituudi kursuste keskkonnas ehk Courses’is [2] ning tööde esitamine ja hindamine käib Moodle’i [3] keskkonnas. Tööde kontrollimine on automatiseeritud, see tähendab, et osalejate töid hindab ning annab neile tagasisidet automaatkontroll. Automaatkontroll eeldab, et esitatud töö vastaks teatud tingimustele ning jätab väga vähe ruumi loominguliste programmidele. Teistes ainetes tihti piisab materjali sirvimisest, siis programmeerimise materjalidega nii ei ole. Programmeerimise materjalid on üldiselt väga sisukad ning iga lause on oluline. Sageli tuleb ühte osa mitu korda lugeda ning materjali kõrval ka koodijuppe katsetada. Kuna sellise materjali läbitöötamisega jäävad mõned osalejad hätta, siis on kursustel olemas ka abiliin, kuhu nad saavad kirju saata. Lisaks materjali läbitöötamisega, tekkis neil ka probleeme automaatkontrolliga, kellel oli mõni muutuja valesti defineeritud, sisendite küsimine vales järjekorras või muu probleem, mida ise ei osanud lahendada. Kuna küsimusi saabus abiliinile hulk kirju (1534 osalejaga kursusel saabus 4 nädala jooksul ~1250 kirja), nähti, et edaspidi peaks sellist olukorda kuidagi leevendama.

Kursuse korraldajad arvasid, et mingil määral võiks kirjade hulka vähendada tõrkeotsija (ingl troubleshooter), mis annab osalejatele samm-sammulisi juhendeid, et ise leida oma probleemile lahendus. Antud bakalaureusetöö eesmärgiks ongi veebirakenduse arendamine, kus saab luua ja kasutada tõrkeotsijaid ning koguda nende kohta statistilist infot. Veebirakendus küll valmib antud kursuste vajadusi silmas pidades, kuid kasutamist sellega ei piirata. See tähendab, et murelahendajaid võib luua igaüks, mis tahes otstarbel.

Kuna sõna tõrkeotsija või veaotsija ei kõla väga hästi, siis sai meie tõrkeotsija nimeks murelahendaja, mis peaks ka osalejatele paremini meelde jääma. Edaspidi antud töös kutsumegi tõrkeotsijaid, mis on loodud valmiva keskkonna abil, murelahendajateks.

Töö autor mõistab loodava keskkonna vajadust, kuna on osalenud mõlema e-kursuse loomisel ning läbiviimisel, samuti on vastanud hulgale osalejate poolt laekunud kirjadele ning on teadlik probleemidest, mis võivad osalejatel tekkida.

Töö esimeses peatükis kirjeldatakse lühidalt, mis on tõrkeotsing ning tuuakse näiteid sarnastest keskkondadest ja lahendustest. Teises peatükis on süsteemi üldine kirjeldus. Neljandas peatükis on kirjutatud kasutatud tehnoloogiatest ja andmebaasi ülesehitusest. Viiendas peatükis on kirjas töö tulemus ning kursusel osalejatelt saadud tagasiside. Lisadena on kaasas rakenduse lähtekood ja selle installeerimise õpetus, keskkonda tutvustav video ning litsents.

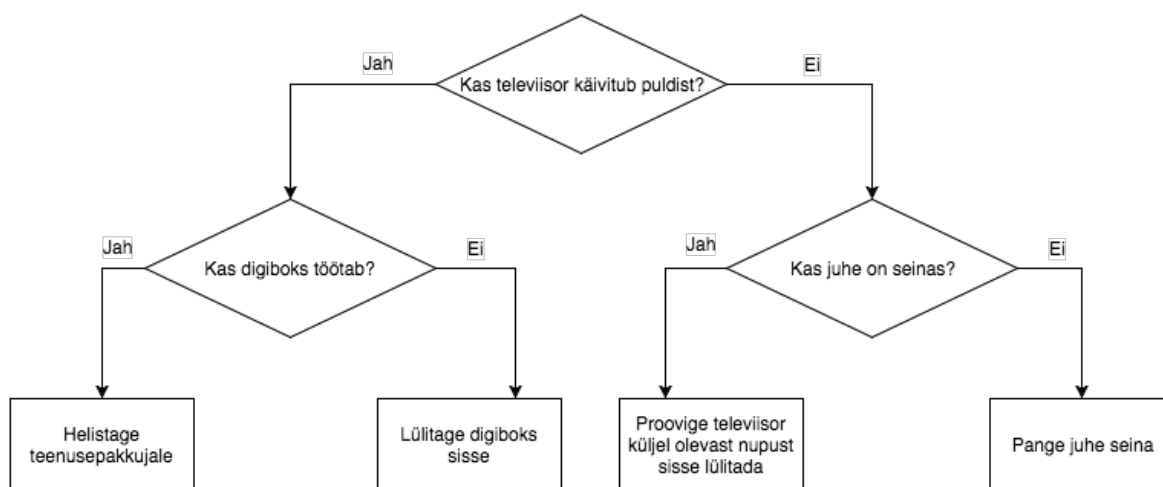
2. Ülevaade

2.1 Tõrkeotsing

David H. Jonassen on välja toonud 11 erinevat probleemi tüüpi [4], antud töös on fookuses neist ainult üks – tõrkeotsing. Tõrkeotsing (ingl troubleshooting) on üks levinumaid probleemide lahendamise viise. Kas parandatakse katkist modemit, keerulist jahutussüsteemi kaubanduskeskuses või kommunikatsiooniprobleeme reklaamiagentuuris – iga sellise probleemi korral üritatakse välja otsida probleemsed kohad ning need välja vahetada või ära parandada, et taastada süsteemi töötav olek [5].

Tõrkeotsijate puhul on kaks erinevat lähenemise viisi: küsimus-vastus ja samm-sammuline lähenemine. Küsimus-vastus lähenemise puhul on iga probleem ja lahendus eraldi välja toodud, kasutajalt ei küsita lisaküsimusi ega viida teda kuhugi edasi. Samm-sammulise lähenemise puhul on tegu protsessiga, kus probleemile või veale otsitakse lahendust loogiliselt ja süsteemselt. Tavaliselt alustatakse kõige üldisema küsimusega ning seejärel liigutakse iga järgneva sammuga üha täpsema probleemi juurde, samal ajal loogiliselt elimineerides mittesobivad variandid. Lõpuks jõutakse kas lahenduseeni või järeldusele, et viga peitub kuskil mujal.

Samm-sammulist lähenemist võib ette kujutada kui puu läbimist, kus alguses on üldised küsimused, mis sügavamale minnes aina täpsemaks lähevad (vt. Joonis 1).



Joonis 1. Samm-sammuline lähenemine puuna.

2.2 Loodava keskkonnaga sarnased keskkonnad

Autor leidis valmiva keskkonnaga 2 sarnast keskkonda: Zingtree [6] ja Yonix [7]. Kuna mõlemad keskkonnad on tasulised, siis kumbagi realselt proovida autoril ei õnnestunud. Mõlemad pakuvad võimalust luua tõrkeotsijaid, mida on võimalik vastavalt kasutaja soovidele ka kujundada. Samuti on mõlemal keskkonnal natuke kallima paketi korral võimalik andmeid analüüsida. Lisaks tõrkeotsijale pakub Zingtree ka võimalust otse kliendiga vestlusi luua.

2.2.1 Hinnavõrdlus ühe kuu raames

Aine "Programmeerimise alused" materjalid on jagatud nädalate kaupa ning iga nädal on kasutusel vähemalt 4 erinevat murelahendajat. Iga murelahendaja saab aktiivsel nädalal umbes 800-1200 vaatamist olenevalt materjali raskusastmest. Seega ühes kuus tehakse ligikaudu $4 * 4 * 1000 = 16000$ klõpsatust.

Keskkond ja pakett	Autoreid ¹	Kuumaks ühe kasutaja kohta	Tasuta kasutuskordi	Kasutuskorra maksumus ²	Kuus kuluv summa
Zingtree - Starter	1	12\$	40	0.25\$	~4002\$
Zingtree - Professional	5	9.8\$	350	0.14\$	~2240\$
Zingtree - Business	10	9.9\$	1000	0.10\$	~1599\$
Zingtree - Corporate	30	9.97\$	4000	0.07\$	~1139\$
Yonyx - Intern	1	6\$	-	0.10\$	~1606\$
Yonyx - Junior	5	11\$	-	0.15\$	~2455\$
Yonyx - Professional	10	17\$	-	0.20\$	~3370\$
Yonyx - Executive	30	22\$	-	0.25\$	~4660\$

¹ Kuna autorite arv on Zingtrees paika pandud, aga Yonixis saab neid suvalise arvu lisada, siis kasutati Yonixi summa arvutamisel samasuguseid autorite arve.

² Kasutuskord - 1 tõrkeotsija külastus, kus tehakse vähemalt üks nupuklõpsatus

2.2.2 Kokkuvõte

On näha, et kõige mõistlikum oleks olnud kasutada Zingtree kõige paremat paketti, kuid ka see oleks olnud üpriski kallis. Seega tundus, et mõistlikum oleks teha oma keskkond, mis arvestab täpselt kursuse korraldajate vajadustega.

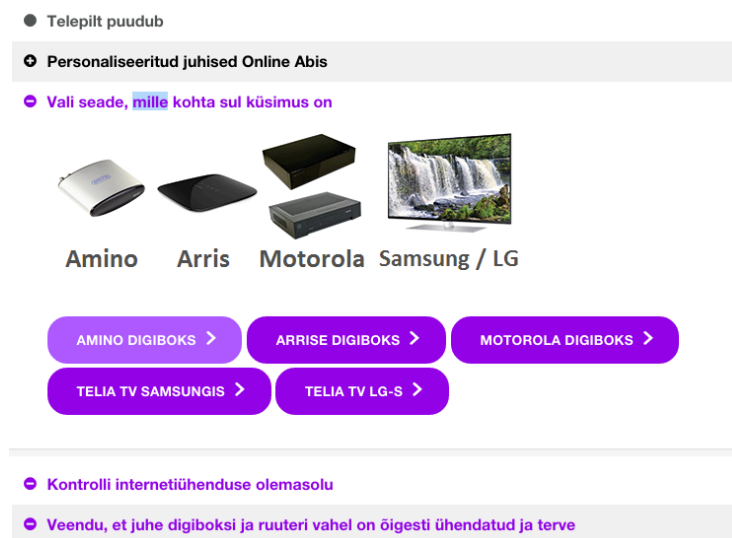
2.3 Tõrkeotsingu näited

Näideteks on võetud 3 keskkonda: Telia, Google ja YouMail. Telia ning Google peaks olema peaaegu kõigile tuttavad ning YouMail peaks olema Eestis suhteliselt

tundmatu keskkond. Vaatleme, mis on antud keskkondadel ühist ning kuidas on nad oma tõrkeotsingu süsteemid üles ehitanud.

2.3.1 Telia

Telia tõrkeotsing [8] on jaotatud alateemadeks, mille sisu on üldiselt küsimus-vastus tüüpi, samas on ka mõned samm-sammulised juhendid (vt. Joonis 2). Iga juhendi lõpus on küsimus, kas juhendist oli abi ning võimalus juhendit kommenteerida.



Joonis 2. Telia tõrkeotsing

2.3.2 Google

Google'i tõrkeotsingu süsteem [9] on sarnane Teliale, kuid palju mahukam, kuna Google'il on tooteharusid tunduvalt rohkem. Alguses on kasutajal valik toodete vahel, mis viib edasi valitud toote enam-levinud probleemide juurde, sealt edasi on üldiselt tegu küsimus-vastus tüüpi abiga. Sarnaselt Teliaga, on ka Google'il iga juhendi lõpus küsimus, kas vastavast juhendist oli abi (vt. Joonis 3).

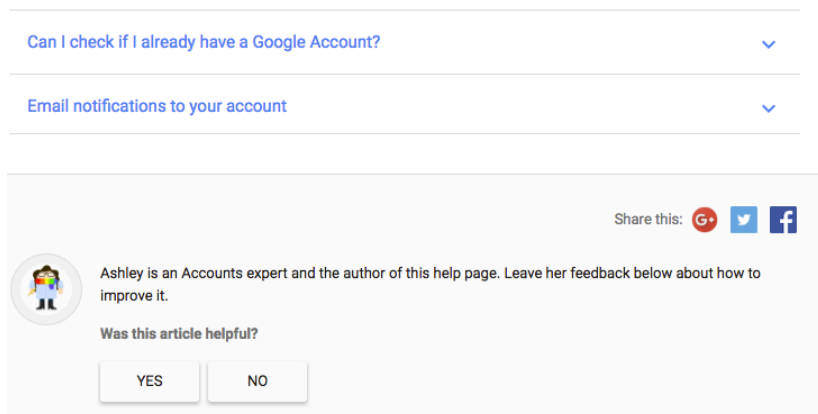
Create a Google Account

A Google Account gives you access to Google products like Gmail, Google+, YouTube, and [more](#) with a single username and password. When you create a Google Account, you can use that same username and password to sign in to any Google product.

Create a Google Account

To create Google Account, go to the [Create your Google Account page](#).

You can also select the **Create an account** link beneath the sign in box in the middle of any Google sign-in page.



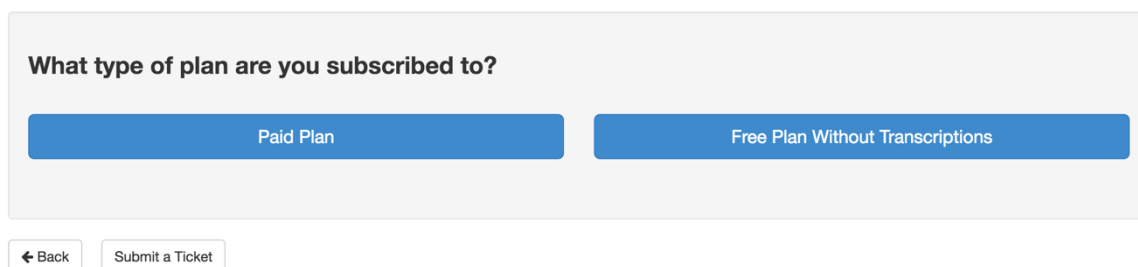
The screenshot shows a feedback section on the Google Account creation page. It includes a profile picture of a person with a backpack, a text box for feedback, and two buttons labeled 'YES' and 'NO'.

Joonis 3. Google'i tõrkeotsing.

2.3.3 YouMail

YouMail pakub teenust, mis vahetab telefoni originaalse kõneposti välja täiustatud versiooni vastu. Abilehel [10] kasutatakse samm-sammulisi juhendeid, osadel sammudel on olemas ka abi küsimise võimalus (Submit a Ticket) (vt. Joonis 4). Samuti mõnede sammude puhul on küsimus, kas juhendist oli kasu.

Type of Plan Question



The screenshot shows a form titled 'What type of plan are you subscribed to?'. It has two buttons: 'Paid Plan' and 'Free Plan Without Transcriptions'. Below the buttons are two smaller buttons: 'Back' and 'Submit a Ticket'.

Joonis 4. YouMaili tõrkeotsing

2.3.4 Kokkuvõte

On näha, et kõik vaadeldud tõrkeotsijad küsivad mingil hetkel külastajalt, kas ta leidis oma probleemile lahenduse. Siit võime järeldada, et andmete kogumine ning analüüsimine on tõrkeotsijate koostajale ülimalt oluline. See võimaldab neid kohendada vastavalt vajadusele, näitab ära, kus kohas on kõige sagedasemad probleemid ning kas juhendist on üldse kasu. Seega oma keskkonda luues peab andmete kogumist ning analüüsimist kindlasti silmas pidama.

3. Keskkonna kirjeldus

3.1 Rollid

Antud keskkonnas on 2 rolli: kasutaja ja külaline. Külaliseks on anonüümne isik, kes tuli oma murele lahendust otsima. Kasutajaks loeme isikut, kes tegeleb murelahendajatega keskkonna siseselt.

3.2 Kuidas külastaja antud keskkonda kasutab?

Külastaja satub antud keskkonda, kui ta klõpsab murelahendaja lingil. Kursuste materjalides asuvad murelahendajatele lingid üldiselt ülesannete juures. Esimesena avaneb külastajale esimese sammu vaade (vt. Joonis 5), kus on mingisuguse probleemi üldine kirjeldus ning valik(ud), mille kaudu külastaja n-ö astub järgmise sammu. Samuti on mõnel sammul olemas nupp “Leidsin lahenduse”, millega külastaja saab murelahendaja loojale anda tagasisidet, et tema murelahendajast oli tõesti abi. Külastaja liikumised kajastuvad ka statistikas.

Kontrollülesanne. Tingimuslause

Mitmed autod hoiatavad võimaliku jää eest, kui temperatuur õues on 4,0 või alla selle.

Koostada programm, mis

- küsib kasutajalt õhutemperatuuri,
- väljastab ekraanile *Ei ole jäätumise ohtu*, kui sisestatu on üle 4,0,
- väljastab *On jäätumise oht*, kui temperatuur on 4,0 või alla selle.

Temperatuuri võib sisestada nii täisarvuna kui kaujukomaarvuna, nt -1.3.

Vajan ülesande lahendamisel abi

© 2016, Tartu Ülikool

Joonis 5. Külastajale avanev vaade [11].

3.3 Mittefunktsionaalsed nõuded

3.3.1 Toetatavus

- Rakendus peab töötama laitmatult järgnevates veebilehitsejates: Firefox, Opera, Google Chrome, Safari, Microsoft Edge ning Internet Explorer (alates versioonist 9+).
- Rakendus peab olema eestikeelse kasutajaliidesega.

3.3.2 Jõudlus

- Süsteem peab suutma teenindada korraga vähemalt 1000 külastajat.
- Server peab vastama soovitatavalt 0.5 sekundi jooksul, maksimaalne vastamise aeg on 2 sekundit.

3.3.3 Käideldavus

- Ükski viga ei tohi süsteemi kokku jooksutada.

- Rakendus peab olema kättesaadav 99% ajast.
- Kriitilised vead tuleb parandada 1 tunni jooksul.

3.3.4 Kasutatavus

- Külastajale avanev kasutajaliides peab olema äärmiselt lihtne.
- Abijuhend peab olema keskkonna siseselt kiiresti kättesaadav.

3.4 Funktsionaalsus

3.4.1 Registreerimine

Registreerimine toimub lehel <http://progtugi.cs.ut.ee/#/login>. Selleks, et alustada registreerimist, klõpsab kasutaja nupul “Uus kasutaja” (vt. Joonis 6), mille peale avaneb kasutajal registreerimise vormi vaade. Registreerimise vormis on järgmised väljad:

- Kasutajatunnus
- Parool
- E-mail
- Eesnimi
- Perekonnanimi

Kui vastavad väljad on täidetud, klõpsab kasutaja nupul “Registreeri”, mille peale toimub kõigepealt kliendipoolne väljade verifitseerimine ning seejärel kontrollitakse serveripoolel, et antud kasutajatunnus ja e-mail oleksid unikaalsed. Kui registreerimine õnnestus, logitakse kasutaja automaatselt sisse ning suunatakse keskkonda. Kui ei õnnestunud, siis server tagastab veateate koodiga 409, mis tähendab, et server sai andmetest aru, kuid nende andmetega polnud võimalik kasutajat registreerida (Conflict) [12].

3.4.2 Sisselogimine

Sisselogimine toimub samuti lehel <http://progtugi.cs.ut.ee/#/login>. Selleks, et logida sisse, sisestab kasutaja oma kasutajatunnuse ning parooli (vt. Joonis 6). Kui sisestati korrektsed väärtused, suunatakse kasutaja automaatselt keskkonda, vastasel juhul tagastab server veateate koodiga 401, mis tähendab, et autentimisel tekkis viga (Unauthorized) [12].

Palun logi sisse

Kasutajanimi

Parool

Logi sisse

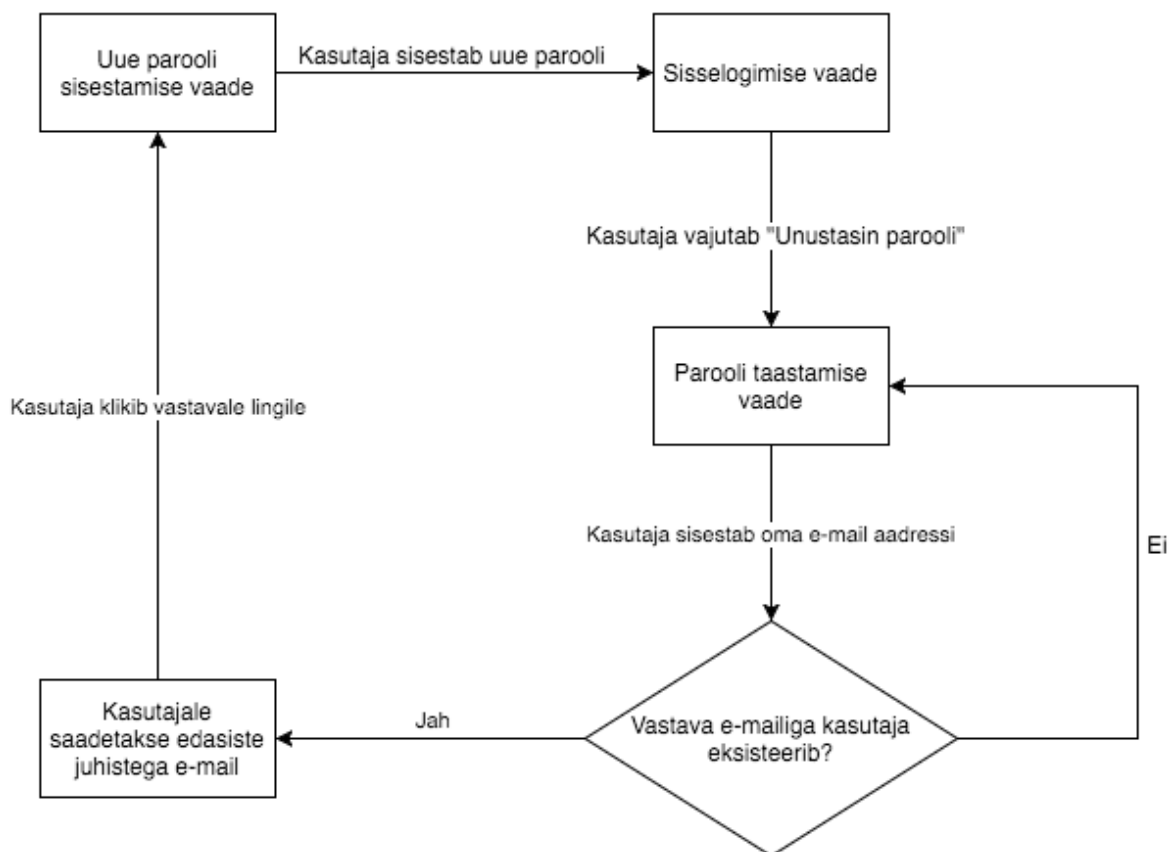
Uus kasutaja

[Unustasin parooli](#)

Joonis 6. Sisselogimine keskkonda.

3.4.3 Parooli taastamine

Kui kasutaja on unustanud oma parooli, siis saab ta taastamise protsessi (vt. Joonis 7) alustada sisselogimise lehelt, klõpsates nupule “Unustasin parooli” (vt. Joonis 6). Seejärel avaneb kasutajale vaade, kus küsitakse temalt e-maili aadressi, mille ta oli sisestanud registreerimise vormis. Kui vastav lahter on täidetud, klõpsab kasutaja nupul “Taasta”, mille peale kontrollitakse serveri poolel, kas vastava e-mailiga kasutaja on süsteemis olemas. Kui vastavat e-maili ei leitud, vastab server koodiga 404 (Not Found) [12] ning kasutajale kuvatakse vastavat veateadet. Kui e-mail oli korrektne, siis saadetakse vastavale e-mailile kiri, kus on kirjas täpsemad juhised ning link parooli taastamiseks. Vastav link kehtib ainult tund aega. Kasutaja klõpsab lingil ning talle avaneb vaade, kus ta saab määrata endale uue parooli. Kui parooliväljad on korrektselt täidetud, klõpsab kasutaja nupul “Salvesta”. Seejärel suunatakse kasutaja tagasi sisselogimise lehele ning samal ajal saadetakse kasutajale ka kinnitus kiri e-mailile, mis ütleb, et parool on edukalt vahetatud.



Joonis 7. Parooli taastamine.

3.4.4 Murelahendaja loomine

Murelahendajate loomine toimub keskkonna põhivaatel, kuhu jõutakse peale sisselogimist. Selleks, et uut murelahendajat luua, klõpsab kasutaja nupul “Lisa murelahendaja”, mis avab murelahendaja loomise dialoogi (vt. Joonis 8), kus ükski väli peale nime pole kohustuslik. Kasutaja täidab vähemalt nimevälja ning klõpsab nupul “Salvesta”, mille peale dialoog sulgub.

Murelahendaja loomine

Nimi

Sildid / märksõnad

Lisa

Lehe Pealkiri

Õnnestumise lehe pealkirja vaikeväärtus

Õnnestumise lehe nupu teksti vaikeväärtus

Õnnestumise lehe sisu vaikeväärtus

B

I

S

¶

¶

¶

Format

Size

A

¶

¶

I_x

¶

¶

Ω

¶

¶

↶

↷

Source

Salvesta

Joonis 8. Murelahendaja loomise dialoog.

3.4.5 Murelahendajate kuvamine

Kõik kasutaja poolt lisatud murelahendajad kuvatakse keskkonna põhivaatel olevas tabelis (vt. Joonis 9).

Murelahendajad	Sildid / koodid	Statistika		Link
Ülesanne 3.1. Suured tähed		Statistika	⚙️ 👤 📄	http://progtugi.cs.ut.ee/ts/56a0b1181f8cb8a227e43015/
Ülesanne 3.1. Vaateratas. Kevad 2016		Statistika	⚙️ 👤 📄	http://progtugi.cs.ut.ee/ts/5706990ba2b8b3bf6e8cba16/
Ülesanne 3.2. Lilled arv. Kevad 2016		Statistika	⚙️ 👤 📄	http://progtugi.cs.ut.ee/ts/5706a722a2b8b3bf6e8cbc31/
Ülesanne 3.2. Paaritute arvude summa		Statistika	⚙️ 👤 📄	http://progtugi.cs.ut.ee/ts/56a160cf1f8cb8a227e4342a/
Ülesanne 3.3. Täring		Statistika	⚙️ 👤 📄	http://progtugi.cs.ut.ee/ts/56a20d071f8cb8a227e434e5/
Ülesanne 3.3. Täring. Kevad 2016		Statistika	⚙️ 👤 📄	http://progtugi.cs.ut.ee/ts/5706acb8a2b8b3bf6e8cbcb9/
Ülesanne 3.4a. Lilled arv v2. Kevad 2016		Statistika	⚙️ 👤 📄	http://progtugi.cs.ut.ee/ts/5706b657a2b8b3bf6e8cbe00/
Ülesanne 3.4a. Summa		Statistika	⚙️ 👤 📄	http://progtugi.cs.ut.ee/ts/56a212341f8cb8a227e43537/
Ülesanne 3.4b. Vabavisked		Statistika	⚙️ 👤 📄	http://progtugi.cs.ut.ee/ts/56a222e01f8cb8a227e43586/
Ülesanne 3.4b. Vabavisked. Kevad 2016		Statistika	⚙️ 👤 📄	http://progtugi.cs.ut.ee/ts/5706c257a2b8b3bf6e8cbfcc/
Ülesanne 3.4c. Male		Statistika	⚙️ 👤 📄	http://progtugi.cs.ut.ee/ts/56a4bd331f8cb8a227e43ec1/
Ülesanne 3.4c. Male. Kevad 2016		Statistika	⚙️ 👤 📄	http://progtugi.cs.ut.ee/ts/5706c7b3a2b8b3bf6e8cc04a/

Joonis 9. Murelahendajate tabel.

13

3.4.6 Murelahendajate seadete muutmine

Kui kasutaja soovib oma murelahendaja seadeid muuta (näiteks lisada silte või muuta nime), siis klõpsab ta vastaval tabeli real hammasratta ikoonile (vt. Joonis 9), mis avab samasuguse dialoogi (vt. Joonis 8) nagu murelahendaja loomiselgi. Kui kasutaja on teinud soovitud muudatused, siis klõpsab ta nupul “Salvesta” ning dialoog sulgub.

3.4.7 Murelahendaja kustutamine

Kui kasutaja soovib murelahendajat kustutada, siis klõpsab ta samuti hammasratta ikoonil (vt. Joonis 9) ning avanenud dialoogi (vt. Joonis 8) jaluses on olemas nupp “Kustuta”. Sellel klõpsates avaneb kinnitamise dialoog, kus kasutajalt küsitakse üle, kas ta on kindel, et soovib antud murelahendaja kustutada. Klõpsates “Kustuta” suletakse mõlemad dialoogid ning murelahendaja kustutatakse. Klõpsates “Tagasi” suletakse kinnitamise dialoog, seadete dialoog jääb lahti.

3.4.8 Murelahendaja jagamine keskkonnasiseselt

Keskkonnasisene jagamine - kasutaja annab teisele kasutajale õiguse oma murelahendajat muuta. Kui kasutaja tahab oma murelahendajat jagada, siis klõpsab ta murelahendajate tabelis vastaval real olevale inimese pea kujuga ikoonile (vt. Joonis 9), seejärel avaneb kasutajale dialoog, kus on võimalik teisi lisada nende e-maili või kasutajanime järgi. Samuti on dialoogis nimekirj kasutajatest, kellele on murelahendajat juba jagatud ning võimalus neid eemaldada. Ainult murelahendaja algsel loojal on õigus oma murelahendajat jagada, teistel vastavat ikooni tabelisse ei ilmu.

3.4.9 Murelahendaja jagamine väljaspool keskkonda

Kui kasutaja on oma murelahendaja valmis saanud ja soovib selle lisada oma leheküljele (näiteks Moodle'isse), siis kopeerib ta murelahendajate tabelist lingi (vt. Joonis 9) ning kleebib selle oma leheküljele. Vastavale lingile klõpsamisel avaneb kasutaja poolt loodud murelahendaja.

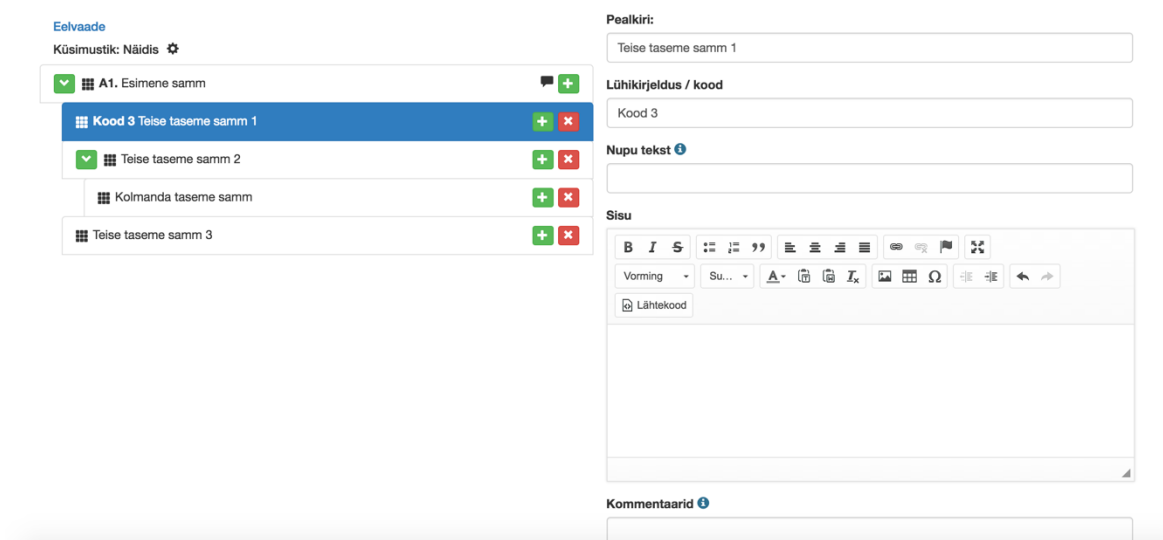
3.4.10 Murelahendaja kloonimine

Kui kasutaja soovib oma murelahendajat kloonida, siis klõpsab ta murelahendajate tabelis olevale kahekordsele dokumendi ikoonile (vt. Joonis 9), mille tulemusena tehakse kogu murelahendaja struktuurist koopia ning vastav murelahendaja ilmub murelahendajate tabelisse. Kloonitud murelahendaja nime lõppu lisatakse “_CLONE”, et kasutajal oleks selgesti arusaadav, milline murelahendaja on kloonitud. Murelahendaja kloonimisel ei kloonita statistika osa.

3.4.11 Murelahendaja sisu loomine

3.4.11.1 Murelahendaja sisu loomise vaade

Selleks, et murelahendajale sisu hakata looma, klõpsab kasutaja murelahendaja tabelis murelahendaja nime peale, mille tulemusel avaneb kasutajale sisu loomise vaade (vt. Joonis 10).



Joonis 10. Murelahendaja sisu loomise vaade.

3.4.11.2 Murelahendaja sammude loomine

Kui kasutaja pole murelahendajasse varem ühtegi sammu lisanud, siis esimese sammu lisamiseks klõpsab kasutaja nupul “Lisa esimene samm”, millele klõpsates tehakse kasutajale esimene samm valmis. Kui kasutaja tahab luua järgnevaid samme, siis klõpsab ta pluss ikooniga rohelisele nupule, mille tulemusena tehakse järgmine samm. Selleks, et kasutajal oleks lihtsam aru saada, kuidas sammud üksteise suhtes asetsevad, on sammude nimekiri puu kujuga. See tähendab, et mida sügavamal samm asub, seda suurema taandega ta kujutatud on. Igale sammule on võimalik lisada järgnevaid samme, mis asetsevad üks aste sügavamal (vt. Joonis 11).



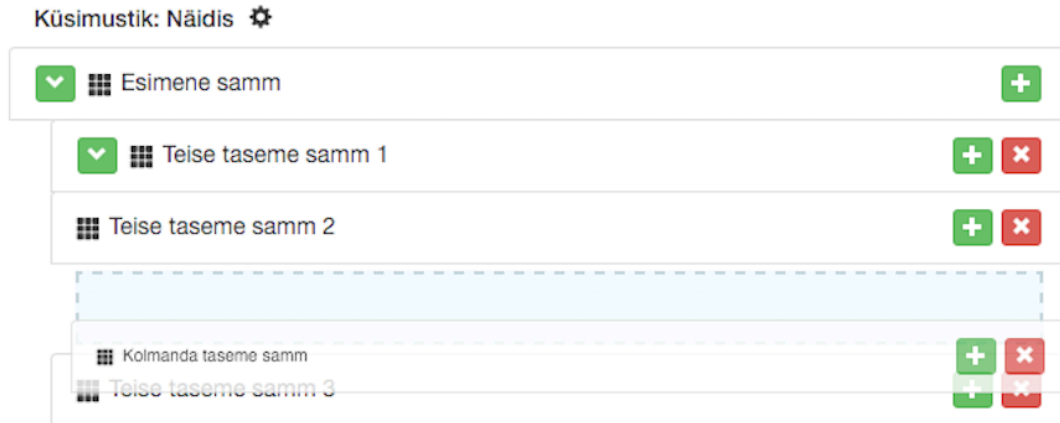
Joonis 11. Murelahendaja sammude nimekiri.

3.4.11.3 Murelahendaja sammude kustutamine

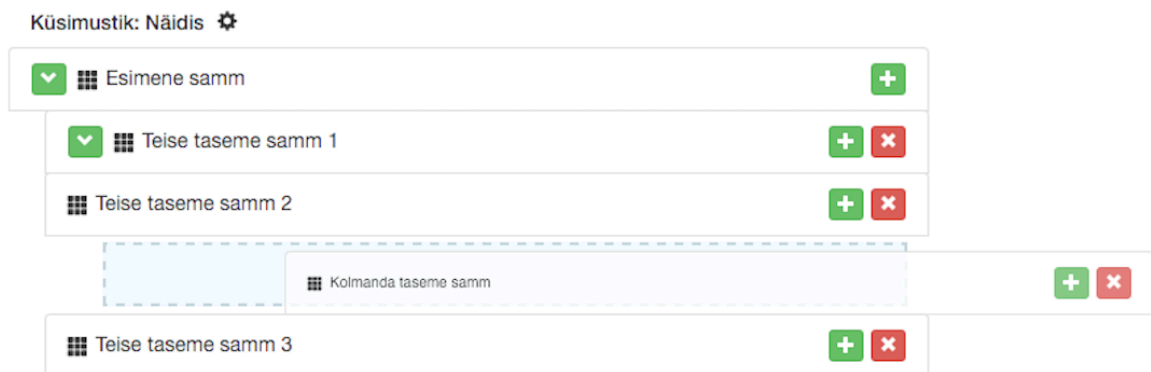
Kui kasutaja soovib teatud sammu kustutada, siis klõpsab ta “x” ikooniga punasele nupule (vt. Joonis 11). Selleks, et vältida kogemata kustutamist, avaneb kasutajal kinnitamise dialoog, kus küsitakse üle, kas ta ikka soovib sammu kustutada. Kui kasutaja klõpsab avanenud dialoogis “Tühista”, siis dialoog sulgub ning midagi rohkem ei juhtu. Kui kasutaja valib “Kustuta”, siis dialoog sulgub ning vastav samm kustutatakse ning samal ajal määratakse alamsammudele uus vanem ehk alamsammud liiguvad üks tase kõrgemale. Esimest sammu ei ole võimalik kustutada.

3.4.11.4 Murelahendaja sammude järjekorra muutmine

Kui kasutaja soovib mingit sammu teise kohta liigutada, siis haarab ta vastava sammu ruudustiku kujuga ikoonist kinni ning tirib sobivale positsioonile. Sügavuse muutmiseks tirib kasutaja sammu kasti vasakule või paremale jälgides taanet (vt. Joonis 12 ja Joonis 13).



Joonis 12. Murelahendaja sammude järjekorra muutmine. Taandeta.



Joonis 13. Murelahendaja sammude järjekorra muutmine. Taandega.

3.4.11.5 Murelahendaja seadete muutmise sisu loomise vaates

Kui kasutaja soovib muuta murelahendaja üldiseid seadeid olles sisu loomise vaatel, siis klõpsab ta murelahendaja nime järel olevale hammasratta ikoonile (vt. Joonis 10). Kasutajale avaneb dialoog, mis on identne keskkonna põhivaates avaneva seadete dialoogiga (vt. Joonis 8).

3.4.11.6 Murelahendaja sammu sisu loomine

Kui kasutaja tahab vastava sammu sisu muuta, klõpsab ta vastava sammu peale, mille tulemusena värvitakse sammu kast siniseks (vt. Joonis 10), et kasutajal teaks, millist sammu ta hetkel muudab ning tal ilmub lehe paremale poole sammu sisu vorm, kus on järgnevad väljad:

- Pealkiri
- Lühikirjeldus
- Nupu tekst
- Sisu

- Kommentaarid
- Märkekastid
 - Tagasi nupp
 - Näita leidsin lahenduse nuppu

Väljad “Lühikirjeldus” ning “Kommentaarid” on keskkonna sisesed abiväljad (vt. Joonis 14), et kasutajal oleks võimalik hiljem paremini orienteeruda või jätta endale ning kaasautoritele mõningaid märkmeid. Lühikirjeldus lisatakse sammude nimekirjas pealkirja ette. Kui sammule lisatakse kommentaare, siis lisatakse ka sammu kastile vastav ikoon juurde. Ülejäänud väljad kuvatakse külastaja vaates (vt. Joonis 15).

Kui kasutaja on märkinud ruudu “Tagasi nupp”, siis kuvatakse talle lisaväli “Tagasi nupu tekst”. Vastav väli on kasutusel kui kasutaja tahab lisaks tavalisele tagasi nupule veel ühe tagasi nupu eraldi valikuna lisada.

Kui kasutaja on märkinud ruudu “Näita leidsin lahenduse nuppu”, siis kuvatakse talle 3 lisavälja: “Õnnestumise lehe pealkiri”, “Õnnestumise nupu tekst”, “Õnnestumise lehe sisu”. Vastavate väljade vaikeväärtusi saab seadistada murelahendaja seadetest (vt. Joonis 8). Antud valik lisab sammule lisa valiku, mille külastaja saab klõpsata kui ta leidis antud sammu juures oma probleemile lahenduse. “Leidsin lahenduse” nupul tehtud klõpsud märgitakse eraldi statistikas.

Joonis 14. Murelahendaja keskkonna sisesed abiväljad.

Pealkiri:
Sissejuhatus. Lisaülesanne. Probleem sisendiga

Lühikirjeldus / kood
1

Nupu tekst [Kuvatakse eelneval sammul valikuna](#)
Vajan ülesande lahendamisel abi

Sisu

Vorming - Su... - A... - I... - L... - Ω - ↩ - ↪

Lähtekood

Kas programmis on ülesande lahendamise juures kasutanud funktsiooni *input* (küside tuleb täpselt selles järjekorras - laiust küpsistes, pikkust küpsistes, kihtide arvu ning küpsiste arv pakis)?

Kommentaariid [?](#)

☐ Tagasi nupp ☒ Näita leidsin lahenduse nuppu? [?](#)

Sissejuhatus. Lisaülesanne. Probleem sisendiga

Kas programmis on ülesande lahendamise juures kasutanud funktsiooni *input* (küside tuleb täpselt selles järjekorras - laiust küpsistes, pikkust küpsistes, kihtide arvu ning küpsiste arv pakis)?

El, kuidas seda teha?

Jah, aga ikka ei tööta

Sain korda

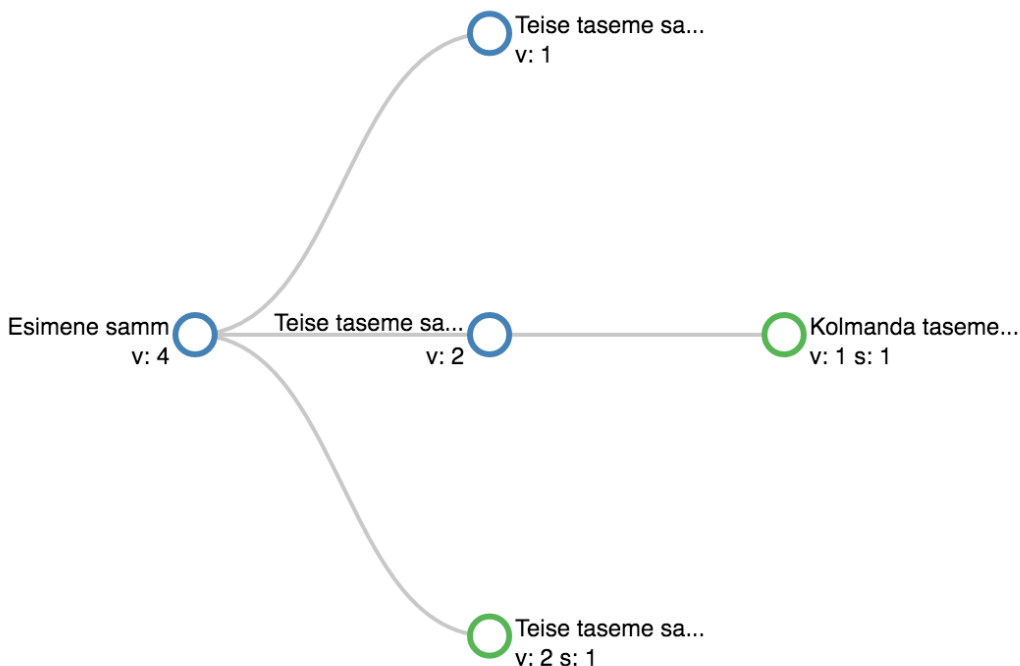
← Tagasi

© 2016, Tartu Ülikool

Joonis 15. Murelahendaja loomise vormi väljade seos lõpptulemusega.

3.4.12 Murelahendaja statistika

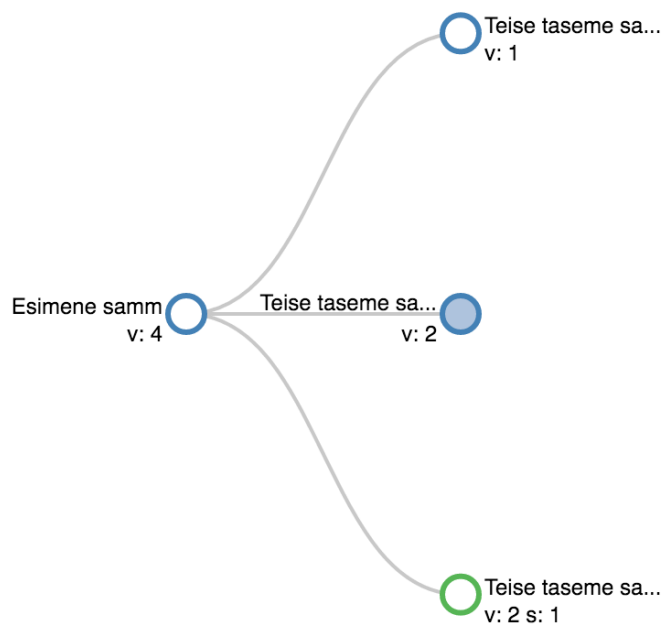
Kui kasutaja tahab vaadata, kuidas tema murelahendajat on kasutatud, klõpsab kasutaja murelahendaja tabelis olevale lingile “Statistika”, mille peale avaneb kasutajale statistika vaade, kus kogu tema murelahendaja ehitus on esitatud puuna (vt. Joonis 16).



Joonis 16. Murelahendaja statistika.

Graafil olevad sinised ringid tähendavad, et sammul puudub “Leidsin lahenduse” valik ning kuvatakse ainult vaatamiste arv. Rohelised ringid tähendavad, et sammul on “Leidsin lahenduse” valik olemas ja lisaks vaatamistele kuvatakse ka arv kui palju inimesi on antud sammul oma probleemile lahenduse leidnud (vt. Joonis 16). Kui kasutaja tahab

puust mingi osa ära peita, siis klõpsab ta vastava sammu ringil, mille tulemusena peidetakse alamsammud ja ringi sisu värv muutub siniseks (vt. Joonis 17).



Joonis 17. Murelahendaja statistika peidetud osaga.

4. Kasutatud tehnoloogiad

Veebikeskkonna serveripoolne osa on realiseeritud kasutades JavaScriptil põhinevat platvormi Node.js [13]. Node.js tuli välja aastal 2009 ning loojaks oli Ryan Dahl. Aastal 2011 tuli välja ka npm (Node Package Manager) [14], mis võimaldab arendajatel väga lihtsalt mooduleid jagada ja hallata. Praeguseks on Node.js kasutajate arvu poolest maailma kiiremini kasvav avatud lähtekoodiga platvorm, millel on praegu juba üle 3,5 miljoni kasutaja ning aastane kasutajate kasv on 100% [15].

Andmebaasiks sai valitud MongoDB [16]. MongoDB on avatud lähtekoodiga dokumentidel põhinev andmebaas, see tähendab, et tegu ei ole relatsioonilise andmebaasiga. Et MongoDB Node.js-iga koos kergemini kasutatav oleks, on kasutusel Mongoose [17] teek, mis tõlgib andmed andmebaasist JavaScripti objektideks.

Veebikeskkonna kliendipoolne osa realiseerimiseks kasutati JavaScripti koos AngularJs [18] raamistikuga, lisaks sellel on kasutusel ka D3.js [19] teek statistika graafikute kuvamiseks ning Bootstrap [20] raamistik dünaamilise kujunduse loomiseks.

Autor valis sellised serveripoolsed tehnoloogiad, kuna polnud nendega varem kordagi kokku puutunud ja tahtis töö käigus ka mõne uue tehnoloogia selgeks saada. Kliendipoolsete tehnoloogiatega oli autor juba tuttav.

4.1 Disain

Tegemist on klient-server rakendusega, kus klient saadab HTTP (Hypertext Transfer Protocol) [21] päringuid serverile, mis vastab kasutades JSON (JavaScript Object Notation) [22] formaati. Projekti alguses oli serveri majutamiseks kasutatud Herokut [23], enne “Programmeerimise alused” pilootkursuse algust kolisime serveri üle Ülikooli serverisse.

4.2 Veebirakenduse serveripoolne osa

4.2.1 Moodulid

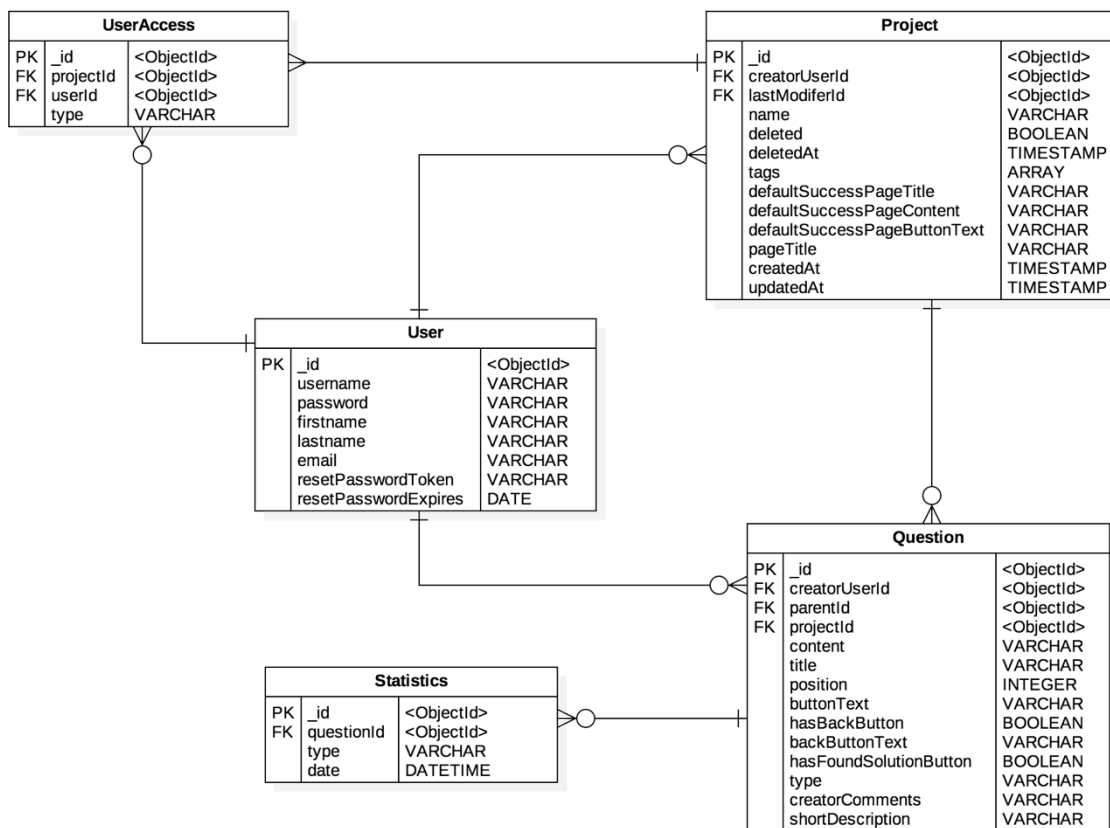
Serveripoolel on kasutusel moodulite haldamiseks kasutusel npm. Lähtekoodis pole ühtegi moodulit, nende nimetused on defineeritud failis package.json koos versiooni numbritega. Selleks, et vajaminevad moodulid kätte saada tuleb käsurealt liikuda lähtekoodi juurkausta ning sisestada käsk “npm install”, mis laeb alla kõik vajaminevad moodulid. Selline lahendus tagab selle, et versioonihaldusesse ei satuks väga palju prügi ning igal arendajal oleks oma arvutiga ühilduvad moodulid. Autor lükkas projekti alguses ka moodulid üles, kuid hiljem tekkisid ühilduvuse probleemid, kuna autor kasutas arendamiseks OS X-i ja live server jooksis Linuxil. Tähtsamad moodulid:

- Mongoose – aitab siduda omavahel Node.js'i ja MongoDB.
- Express – marsruutimine (ingl routing).
- JSON Web Token – kompaktne ja turvaline moodus osapoolte vahel JSON formaadis informatsiooni vahetada.
- Moment.js – lihtsustab töötamist kuupäevade ja kellaaegadega .
- Nodemailer – e-mailide saatmine Node.js'iga.
- Passport – lihtsustab autentimist.

4.2.2 Andmebaas

Andmebaasiks sai valitud MongoDB. Selle sidumine Node.js'iga on üpriski lihtne tänu Mongoose'i teegile. Andmemudel (vt. Joonis 18) on kokku 5 dokumenti:

- **User** – selles dokumendis on kirjas veebirakenduse kasutajate andmed.
- **UserAccess** – selles dokumendis on kirjas, millistele murelahendajatele on mingil kasutajal ligipääs.
- **Project** – selles dokumendis on kirjas murelahendaja üldine info.
- **Question** – selles dokumendis on kirjas kogu sammuga seonduv info
- **Statistics** – selles dokumendis on kirjas iga sammu vaatamised koos aegadega.



Joonis 18. Andmemudel.

Kuna autor pole veel refaktoriseerimisega tegeleda jõudnud, siis on kahe dokumendi nimi hetkel natukene halb saanud. Tulevikus on autoril plaanis dokumendi “Project” nimi muuta ümber “Troubleshooter”-iks ning dokumendi “Question” nimi muuta ümber “Step”-iks.

4.2.3 E-mailide saatmine

E-mailide saatmine käib läbi teenusepakkuja SendGrid [24], mis on serveri poolel ära ühendatud Nodemailer mooduli abil. E-mailide saatmine on kasutusel parooli taastamisel.

4.3 Kliendipoolne osa

Veebileheküljed on tehtud HTML (HyperText Markup Language) ja CSS (Cascading Style Sheets) abil. Interaktiivsuse eest hoolitseb skriptikeel JavaScript koos AngularJs raamistikuga. Kujundusel on abiks Bootstrapi raamistik.

4.3.1 AngularJs

AngularJs on struktuurne raamistik, mis on loodud dünaamiliste veebirakenduste jaoks. See võimaldab luua malle kasutades HTML-i ning samuti laiendada HTML-i süntaksit, et koostada oma rakenduse komponente selgelt ja lühidalt. Samuti AngularJsi lihtne andmete sidumine vähendab kõvasti koodi hulka, mida muidu peaks kirjutama [25].

Kasutusele sai AngularJs võetud mitmel põhjusel. Esiteks oli autoril sellega varasem tööalane kogemus juba olemas. Teiseks on sellega väga mugav andmeid kuvada ning uuendada. Samuti on koodi hulk tunduvalt väiksem kui oleks kasutanud mingit muud raamistikku. Näiteks statistika vaate HTML koosneb ainult 3-st reast, ülejäänud osa teeb ära direktiiv. Eriti meeldib autorile HTML süntaksi laiendamine ehk mingile DOM (Document Object Model) elemendile erilise käitumine lisamine.

4.3.2 Graafikute kuvamine

Statistika vaates kuvatavad graafikud on joonistatud D3.js teegi abil ning aluseks on võetud Mike Bostocki Collapsible Tree [26] kood, mis on ümber kohandatud AngularJsi direktiiviks.

5. Tulemus

5.1 Kasutegur

MOOCidel osalejate kirjade analüüsimisega ning murelahendajate koostamisega tegeles Kaspar Hollo. Oma bakalaureusetöös võrdles ta sügis- ja kevadsemestril toimunud MOOCide “Programmeerimisest maalähedaselt” laekunud kirjade arvu ning leidis, et tänu murelahendajale vähenes nende arv märkimisväärselt. Samuti on ta oma töös välja toonud, et murelahendajate loomise keskkond on ajaga läinud aina paremaks [27].

5.2 Tagasiside

Igal kursusel küsitakse osalejatelt iga kahe nädala tagant tagasisidet, milles on ka küsimused murelahendajate kohta. Üldiselt kommenteerisid osalejad murelahendajate sisu, mitte välimust ega kasutamist. Hetkel käimasoleva “Programmeerimise alused” e-kursuse viienda ja kuuenda nädala tagasisideküsitluses [28] paluti osalejatel vastata sellisele küsimusele: “Käesoleval kursusel on meil kasutusel ülesannete juures murelahendajad. Palun avaldage arvamust, kas ja milliste ülesannete puhul Teie olete seda juba kasutanud ning kas olete abi saanud?” Autor toob välja mõned saabunud vastused:

“Mitme ülesande puhul on olnud murelahendaja abiks. Tihtipeale jääbki midagi kahe silma vahele ning siis on murelahendajast kasu. Kohe meenub Kontrollülesanne 5.4a - Reisidiilid, kus ei olnud mul programmifail ja tekstifail samas kaustas, murelahendajast oli koheselt abi.”

“Murelahendaja võimaldab ise ülesannete kallal pusida. 6 nädala jooksul olen vaid kahe ülesande korral meili teel abi küsinud. Igati hea abiline.”

“Olen kasutanud. Abi ei ole eriti saanud. Minu mure oli ilmselt mujal”

“5. ja 6. nädala teema juures ma murelahendajat ei kasutanud, kuna ülesanded olid paraja keerukusega ning leidsin automaatkontrolli abil üles ka need mõned vead, mis ma eisalgu ülesandeid lahendades tegin. Varasemalt olen siiski murelahendajat kasutanud ning leian, et see on vajalik ja väga hea viis võimalike vigade tuvastamisel ja lahenduste leidmisel.”

6. Kokkuvõte

Antud bakalaureusetöö käigus valmis murelahendajate keskkond. See võimaldab sisseloginud kasutajatel luua, muuta, kopeerida ning jagada murelahendajaid. Samuti on kasutajal olemas ülevaade, kuidas tema murelahendajaid on kasutatud ning kui palju külastajaid on enda probleemile lahenduse leidnud. Kõik algselt plaanitud funktsioonid on keskkonnas olemas ning keskkond on juba mitmel kursusel ka kasutusel olnud. Autoril on ka plaanis keskkonda edasi arendada, et muuta seda veel kasutajasõbralikumaks, kiiremaks ning parendada statistiliste andmete kogumist.

Kaspar Hollo bakalaureusetöös selgus, et murelahendajad muudavad MOOCi efektiivsemaks, seda näitavad laekuvate kirjade arvu vähenemine (enam-vähem samade osalejate arvude puhul saadeti ilma murelahendajata kursusel kokku 1250 kirja, murelahendajaga kursusel 750) ning murelahendaja keskkonna kaudu kogutud andmed, mille kohaselt saadi 1430 osalejaga kursusel “Programmeerimine maalähedaselt” abi 2180 korral [27].

7. Viited

[1] MOOC.

<https://et.wikipedia.org/wiki/MOOC/> (02.05.2016)

[2] Courses.

<https://courses.cs.ut.ee/> (02.05.2016)

[3] TÜ Moodle.

<https://moodle.ut.ee/> (02.05.2016)

[4] David H. Jonassen, 2010, "Learning to Solve Problems: A Handbook for Designing Problem-Solving Learning Environments", Routledge, 2010, lk 11. [Online].

https://books.google.ee/books?hl=en&lr=&id=DrwuCgAAQBAJ&oi=fnd&pg=PP1&ots=nlt51McQd9&sig=BCHJi7bbzzCK2gpLlAjnOS5xix4&redir_esc=y#v=onepage&q&f=false (02.05.2016)

[5] D.H. Jonassen, W.Hung, 2006, "Learning to Troubleshoot: A New Theory-Based Design Architecture", Springer US, lk 78. [Online].

<http://link.springer.com/article/10.1007/s10648-006-9001-8> (02.05.2016)

[6] Zingtree.

<http://zingtree.com/> (02.05.2016)

[7] Yonyx.

<http://corp.yonyx.com/> (02.05.2016)

[8] Telia tõrkeotsing.

<https://www.telia.ee/abi/> (02.05.2016)

[9] Google'i tõrkeotsing.

<https://support.google.com/> (02.05.2016)

[10] YouMaili tõrkeotsing .

<https://support.youmail.com/hc/en-us> (02.05.2016)

[11] Murelahendaja näide

<http://progtugi.cs.ut.ee/#/ts/56fda645a2b8b3bf6e8c2493/> (02.05.2016)

[12] HTTP Status Codes.

http://www.restpatterns.org/HTTP_Status_Codes (02.05.2016)

[13] Node.js.

<https://nodejs.org/en/about/> (02.05.2016)

[14] npm.

<https://www.npmjs.com/about> (02.05.2016)

[15] World's Fastest Growing Open Source Platform Pushes Out New Release.

<https://nodejs.org/en/blog/announcements/v6-release/> (02.05.2016)

- [16] MongoDB.
<https://www.mongodb.org/> (02.05.2016)
- [17] Mongoose.
<http://mongoosejs.com/> [02.05.2016]
- [18] AngularJs.
<https://angularjs.org/> (02.05.2016)
- [19] D3.js.
<https://d3js.org/> (02.05.2016)
- [20] Bootstrap.
<http://getbootstrap.com/> (02.05.2016)
- [21] HTTP.
https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol (02.05.2016)
- [22] JSON.
<http://www.json.org/> (02.05.2016)
- [23] Heroku.
<https://www.heroku.com/> (02.05.2016)
- [24] SendGrid.
<https://sendgrid.com/> (02.05.2016)
- [25] What Is Angular?
<https://docs.angularjs.org/guide/introduction> (02.05.2016)
- [26] Mike Bostock - Collapsible Tree.
<http://bl.ocks.org/mbostock/4339083> (02.05.2016)
- [27] Kaspar Hollo, Programmeerimise e-kursusel osalejate küsimuste analüüs ja selle põhjal murelahendajate koostamine, 2016, Tartu Ülikooli arvutiteaduse instituut, bakalaureusetöö.
- [28] Viienda ja kuuenda nädala tagasiside kursusele “Programmeerimise alused”.
<https://docs.google.com/forms/d/180vi9XE4dVHzvRyPsIvcBdjIp-VOV33kHGkmFI6EBrE/viewform> (02.05.2016)

I. Rakenduse üles seadmine Linux masinal

Rakenduse kood on kokkupakitud ja lisana kaasas.

Eeldused:

- Arvutisse on installeeritud Node.js ja npm.
- Arvutis võiks joosta MongoDB (kui ei ole soovi oma andmebaasi kasutada, saab ka Herokus olevat test andmebaasi kasutada).

Pakkige rakenduse lähtekood lahti ning liikuge käsureal rakenduse kausta ning andke käsk “npm install”, mis installeerib automaatselt kõik vajaminevad moodulid.

Rakenduse kaustas on fail **env.js**. Kui on soov enda andmebaasi jooksutada, siis tuleks sinna lisada andmebaasi URI, kui jätta URI tühjaks (`process.env['MONGODB_URI'] = ''`), siis kasutatakse automaatselt Herokus olevat test andmebaasi. Secret token on kasutusel JWT (JSON Web Token) genereerimisel. SendGrid on kasutusel ainult paroolide taastamisel, selle võib alguses vahele jätta.

Seejärel andke käsureal käsk “node server.js”, mis käivitab rakenduse. Kui kõik läks hästi on rakendus kättesaadav aadressil localhost:8080.

II. Demo video

Rakenduse lähtekoodiga on kaasas ka demo video “Murelahendaja demo.mp4”, kus kirjeldatakse keskkonda ning tehakse üks näidis murelahendaja.

III. Litsents

Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks

Mina, **Vello Vaherpuu**,
(*autori nimi*)

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) enda loodud teose
Murelahendajate loomise keskkond,
(*lõputöö pealkiri*)

mille juhendaja on Eno Tõnisson,
(*juhendaja nimi*)

- 1.1.reprodutseerimiseks säilitamise ja üldsusele kättesaadavaks tegemise eesmärgil, sealhulgas digitaalarhiivi DSpace-is lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
- 1.2.üldsusele kättesaadavaks tegemiseks Tartu Ülikooli veebikeskkonna kaudu, sealhulgas digitaalarhiivi DSpace'i kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.
2. olen teadlik, et punktis 1 nimetatud õigused jäävad alles ka autorile.
3. kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest tulenevaid õigusi.

Tartus, **12.05.16**